Lib Curl

Table of Contents

1. Introduction

libcurl - the multiprotocol file transfer library

Official curl / libcurl site

2. CookBook

2.1. Using LibCurl with Visual Studio

Interesting article on Stackoveflow :

Intall properly libcurl for visual studio

In my case I used

- libcurl_a.lib
- crypt32.lib
- Wldap32.lib
- Normaliz.lib

2.1.1. Practical installation

Version downloaded: **curl-7.83.1.zip**

Steps :

1. Download version
2. Unzip
3. Open Visual Studio 2022 Developer Command Prompt v17.0.4
4. Go to build folder D:\ccp_vhdd_app*curl-7.83.1\winbuild*
5. Create static library by running the command

```
nmake /f Makefile.vc mode=static
```

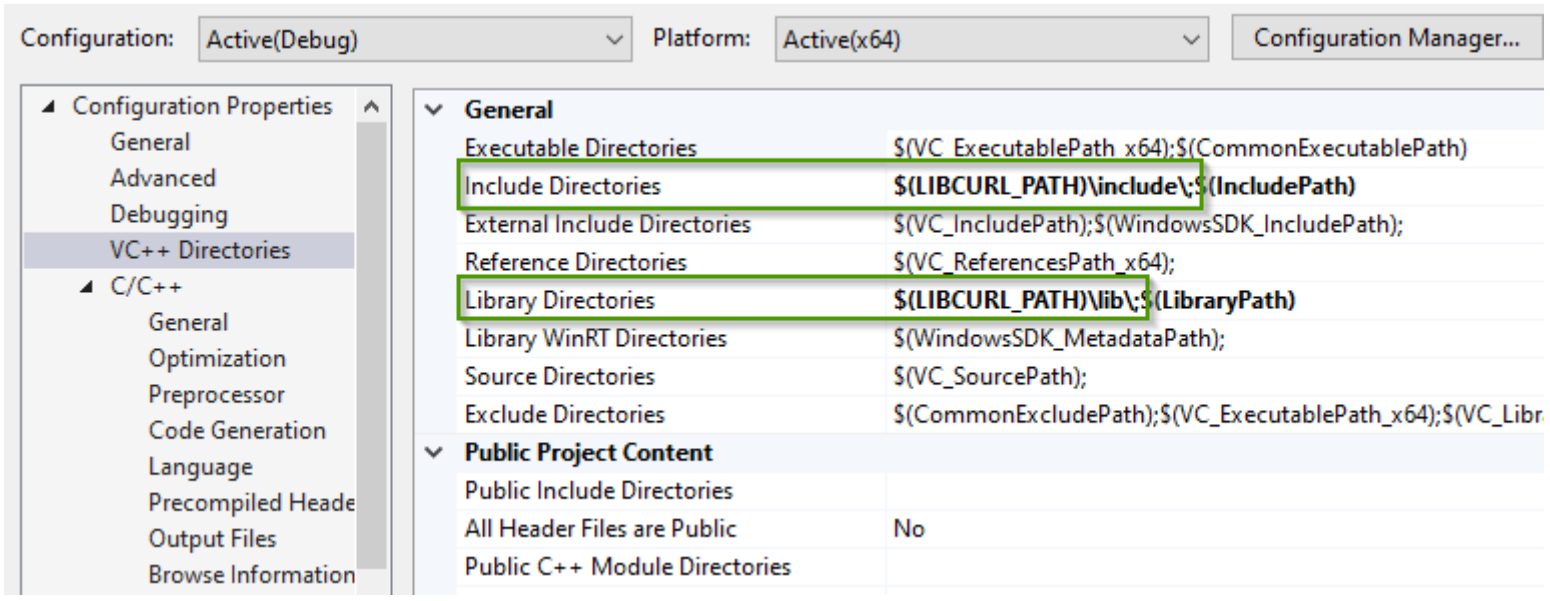
6. This will build curl as a static library into

```
D:\ccp_vhdd_app\curl-7.83.1\builds\libcurl-vc-x64-release-static-ipv6-sspi-schannel
```

7. Definie environment variable that points to the static build

```
$env:LIBCURL_PATH = 'D:\ccp_vhdd_app\curl-7.83.1\builds\libcurl-vc-x64-release-static-ipv6-sspi-schannel'
```

8. Set Visual Studio environment
   a. Add libcurl in Include Directories + Library Directories

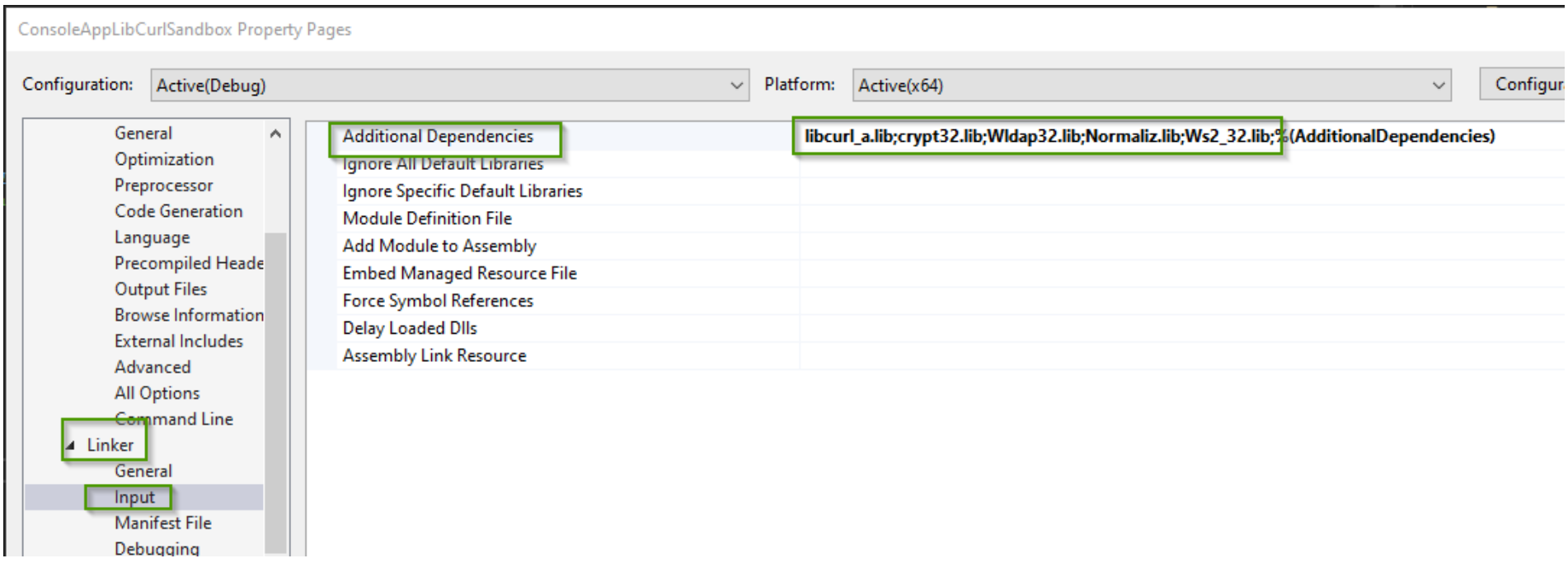


or in project file (*.vcxproj file)

```
<PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|x64'">
  <IncludePath>$(LIBCURL_PATH)\include\;$(IncludePath)</IncludePath>
  <LibraryPath>$(LIBCURL_PATH)\lib\;$(LibraryPath)</LibraryPath>
</PropertyGroup>
```

b. Add aditional libraries for the linker

```
libcurl_a.lib
crypt32.lib
Wldap32.lib
Normaliz.lib
Ws2_32.lib
```



or in project file (*.vcxproj file)

```
  <ItemDefinitionGroup Condition="'$(Configuration)|$(Platform)'=='Debug|x64'">
    <ClCompile>
      <WarningLevel>Level3</WarningLevel>
      <SDLCheck>true</SDLCheck>
      <PreprocessorDefinitions>_DEBUG;_CONSOLE;%(PreprocessorDefinitions)</PreprocessorDefinitions>
      <ConformanceMode>true</ConformanceMode>
    </ClCompile>
    <Link>
      <SubSystem>Console</SubSystem>
      <GenerateDebugInformation>true</GenerateDebugInformation>
      <AdditionalDependencies>
libcurl_a.lib;crypt32.lib;Wldap32.lib;Normaliz.lib;Ws2_32.lib;%(AdditionalDependencies)</AdditionalDependencies>
    </Link>
  </ItemDefinitionGroup>
```

c. The definition CURL_STATICLIB should be added to the project. See below example of minimal source code in C++

```cpp
#include <iostream>
#define CURL_STATICLIB
#include <curl/curl.h>

int main()
{
    CURL* curl;
    CURLcode res;

    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, "https://www.google.com");
        /* Perform the request, res will get the return code */
        res = curl_easy_perform(curl);
        /* Check for errors */
        if (res != CURLE_OK)
            std::cerr << "curl_easy_perform() failed:" << curl_easy_strerror(res) << std::endl;


        /* always cleanup */
        curl_easy_cleanup(curl);
    }
}
```

3. Development with C++

3.1. Introduction

Libcurl with c There's basically only one thing to keep in mind when using C instead of C when interfacing libcurl:

The callbacks CANNOT be non-static class member functions

Example C++ code:

```cpp
class AClass {
static size_t write_data(void *ptr, size_t size, size_t nmemb, void *ourpointer)
    {
        /* do what you want with the data */
    }
}
```

3.2. Exemples

3.2.1. Simple get with writting to the file

```
#include <iostream>
#define CURL_STATICLIB
#include <curl/curl.h>

// to avoid issue with fopen
#pragma warning(disable:4996)

int main()
{
    CURL* curl;
    CURLcode res;

    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, "https://www.google.com");

        FILE* file = fopen("google_com.htm", "w");

        curl_easy_setopt(curl, CURLOPT_WRITEDATA, file);
        /* Perform the request, res will get the return code */
        res = curl_easy_perform(curl);
        /* Check for errors */
        if (res != CURLE_OK)
            std::cerr << "curl_easy_perform() failed:" << curl_easy_strerror(res) << std::endl;


        /* always cleanup */
        curl_easy_cleanup(curl);
        fclose(file);
    }
}
```